

SatelliteNet: Satellite Image Smoke Detection

Kshitij Bhat, Mihir Gadgil

A project report submitted in partial fulfillment of the
requirements for

DATS 6501: Data Science Capstone

Dr. Abdi Awl

The George Washington University

April 2020

Contents

1	Introduction	2
2	Literature Review	3
3	Dataset	4
4	Methodology	5
4.1	Spatial Attention	5
4.2	Channel Attention	6
4.3	SatelliteNet Structure	7
4.4	Training	7
5	Implementation	9
6	Results	10
7	Conclusion	14
8	Limitations and Future Work	15
	References	16

1 Introduction

Fires, whether man-made (arson, explosion, etc) or natural (wildfires, volcanic eruption, etc) cause damages to property, domestic, and/or wildlife on a large scale. The velocity of spread, the extent of damage, and the duration of the fire depends upon various factors. It is always better to detect the fire as early as possible and take necessary countermeasures to mitigate the fire hazard.

While the small scale fires like housefires, can be detected early by the smoke detectors installed in the structures, the early detection of large scale fires, precisely, wildfires yet a big unsolved challenge to date. One of the approaches is to capture and identify the fire in satellite images in real-time. But identifying fire through satellite has its challenges. For instance, fire could only be detected by satellite view if it is spread over a large area and burning at considerably high temperature, or fires can also be concealed by the clouds and would not be captured in a satellite image. These challenges can be overcome by identifying smoke in the satellite images. Smoke rises to higher altitudes, and covers larger areas even when the fire is localized.

One of the existing researches and models proposed for identifying smoke in satellite images is SmokeNet [1], which proposes use of channel and spatial attention calculation [2]. The network proposed in SmokeNet is complex and creates a large number of parameters that require high computing power to execute. Our project aims at simplifying the model and achieving similar or better results than the existing network.

We will start by reviewing the existing methods of smoke detection in

the next section. It will be followed by describing some concepts used in our model. Finally we will discuss and summarise our results, and discuss possible avenues for further improvement in the future.

2 Literature Review

Varying shapes, colors, spectral properties, and similarities to other atmospheric phenomena make it a challenging task to detect smoke in satellite images. The atmospheric phenomena that resemble smoke the most are clouds, haze, and dust. Apart from these, land features and to some extent water surfaces can also be confused for smoke. One popular method of smoke detection in images is manual visual interpretation. This is done using either a true-color image or a false-color image, created using three spectral bands from satellite imagery. Of course, this method cannot be used to process large amounts of data.

For large scale needs, usually a method based on multiple thresholds is used. Thresholds are generated for reflectance or brightness temperature for specific spectral bands. These thresholds are optimized using historical data to best exclude clouds and other above-mentioned phenomena. This method works well for the locality it is developed for, but is impractical to scale. Geographical and temporal variations make it difficult to find optimal thresholds. Sub-optimal thresholds mean a higher possibility of missing a fire in some areas and that is undesirable.

Yet other studies have explored using K-means clustering to detect smoke areas. Neural networks have also been studied to identify pixels

with smoke. The classification labels in these studies, like clouds, water, and vegetation, have been a bit simplistic compared to reality.

A neural network architecture proposed by [1] attempts to detect smoke at the image level rather than pixel level. The model, called SmokeNet, is a convolutional neural network. It uses residual learning proposed by [3] and spatial and channel attention mechanisms. We will discuss the mathematical details of these mechanisms in the next section.

3 Dataset

The source of the dataset is the UNCC data repository [4]. The dataset has six wildfire-related scene classes based on the data of MODIS sensor mounted on Terra and Aqua satellites. The six classes are Cloud, Dust, Haze, Land, Seaside and Smoke. Smoke is the main attribute of determining fire and can quickly spread over a larger area. The dataset consists of images of smoke in different diffusion ranges. Identifying smoke helps early detection of large scale fires.

Cloud, Dust and Haze are considered to be the most competing classes with Smoke because of the texture and spectral similarities. The images of Cloud captured over a water body appears to be greyscale and hence pose challenges in classification. Images of Land and Seaside are included to help learn the differences of various terrains and blue backgrounds of seaside which in turn help in classification. The process of manual acquisition of images and labeling has been explained in the associated research paper [1].

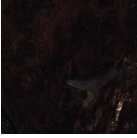
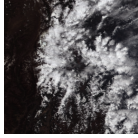




Smoke	Cloud	Dust	Haze	Land	Seaside
1016	1164	1009	1002	1027	1007
					

Table 1: Distribution of classes in the dataset and a sample of each class.

4 Methodology

The network used by this study (henceforth referred to as SatelliteNet) follows the pattern of SmokeNet but excludes the multiple fully connected layers and thereby considerably reduces the number of trainable parameters of the network.

Our goal was to recreate the spatial and channel attention mechanisms from the SmokeNet paper. But the mathematics described in the paper did not make sense and their source code was not available either. We could have tried to reverse engineer the mathematics from the code if it was available. This is how we interpreted the mathematical details using [2] and [5] and it is what we have implemented.

4.1 Spatial Attention

Spatial attention works by combining all channels of the input into a single matrix of the same spatial shape as the input. The output of spatial attention is the elementwise multiplication of this matrix with the input. In this way, the matrix acts like a weight for every pixel location of the input and hence the name, spatial attention. Mathematically, it is described as

follows:

Consider an input $V \in \mathbb{R}^{C \times H \times W}$, where C , H and W are number of channels, height and width respectively. We begin by flattening each channel, $U \in \mathbb{R}^{C \times L}$ where $L = H \cdot W$.

$$V' = f_2 (W_2 * f_1 (W_1 * U + b_1) + b_2) \quad (1)$$

where $*$ denotes the convolution operation, f_1 is the relu activation function, f_2 is the sigmoid activation function, $W_1 \in \mathbb{R}^{C/r \times C \times 1}$, $b_1 \in \mathbb{R}^{C/r}$, $W_2 \in \mathbb{R}^{1 \times C/r \times 1}$, $b_2 \in \mathbb{R}^1$ and $V' \in \mathbb{R}^{1 \times L}$. r is called reduction ratio, which can be treated as a hyperparameter of the model.

$$P = V' \cdot U \quad (2)$$

where the multiplication elementwise across channels and $P \in \mathbb{R}^{C \times L}$. P , the output of spatial attention is reshaped back into $C \times H \times W$ shape.

4.2 Channel Attention

Channel attention, unlike spatial attention, combines all pixel locations into a single scalar for every channel. This gives us a C dimensional vector where C is the number of channels in the input. Every channel of the input is multiplied by the corresponding element of the attention vector. In this way, the vector acts like a weight for every channel of the input and hence the name, channel attention. Mathematically, it is described as follows:

Consider an input V , same as in the spatial attention section. We begin by flattening each channel as before into U . The attention vector is obtained

using average pooling with a kernel of size L (global average pooling). Let's call the resulting vector $U' \in \mathbb{R}^C$.

$$V'' = f_2 (W_2' \cdot f_1 (W_1' \cdot U' + b_1') + b_2') \quad (3)$$

where f_1 and f_2 are as before, $W_1' \in \mathbb{R}^{C/r \times C}$, $b_1' \in \mathbb{R}^{C/r}$, $W_2' \in \mathbb{R}^{C \times C/r}$, $b_2' \in \mathbb{R}^C$ and $V'' \in \mathbb{R}^C$. r is the same as before.

$$Q = V'' \cdot U \quad (4)$$

where the multiplication is elementwise across the spatial dimensions and $Q \in \mathbb{R}^{C \times L}$. Q , the output of channel attention is reshaped into $C \times H \times W$ shape.

4.3 SatelliteNet Structure

A residual block, followed by one or more of spatial and channel attention forms the basic building block of the Residual Attention Module (see SmokeNet paper for structure). The RA module downsamples and upsamples the input to concentrate and extract features. See table 2 for comparison between SmokeNet and SatelliteNet structures.

4.4 Training

Initially we followed the training method from SmokeNet as closely as possible. But the preliminary results showed us that the model's performance would be subpar in that case.

Output Size	SmokeNet	SatelliteNet
$64 \times 112 \times 112$	conv 7×7 , stride 2, padding 3	
$64 \times 56 \times 56$	max pool 3×3 , stride 2, padding 1	
$256 \times 56 \times 56$	conv 1×1 , 64	
	conv 3×3 , 64	conv 1×1 , 64, batch norm
	conv 1×1 , 256	conv 3×3 , padding 1, 64, batch norm
	2fc (16, 256)	conv 1×1 , 256, batch norm
	2fc (16, 256)	RA module - 3
	RA-SC module1	max pool 2×2 , stride 2
$512 \times 28 \times 28$	conv 1×1 , 128	
	conv 3×3 , 128	conv 1×1 , 128, batch norm
	conv 1×1 , 512	conv 3×3 , padding 1, 128, batch norm
	2fc (32, 512)	conv 1×1 , 512, batch norm
	2fc (32, 512)	RA module - 2
	RA-SC module2	max pool 2×2 , stride 2
$1024 \times 14 \times 14$	conv 1×1 , 256	
	conv 3×3 , 256	conv 1×1 , 256, batch norm
	conv 1×1 , 1024	conv 3×3 , padding 1, 256, batch norm
	2fc (64, 1024)	conv 1×1 , 1024, batch norm
	2fc (64, 1024)	RA module - 1
	RA-SC module3	max pool 2×2 , stride 2
$2048 \times 7 \times 7$	conv 1×1 , 512	
	conv 3×3 , 512	conv 1×1 , 512, batch norm
	conv 1×1 , 2048	conv 3×3 , padding 1, 512, batch norm
	2fc (128, 2048)	conv 1×1 , 2048, batch norm
	2fc (128, 2048)	
	This block $\times 3$	
2048	average pool 7×7 , stride 1	
6	fc 6×2048 , softmax	

Table 2: Structure comparison of SmokeNet and SatelliteNet. The numbers after convolution denote the kernel size and number of output channels. The number after RA module in SatelliteNet denotes how many times input gets downsampled and upsampled.

To review, SmokeNet implemented their methodology using PyTorch. They used Adam optimizer with weight decay 10^{-4} . Initial learning rate was set to 10^{-3} , decreased by a factor of 0.5 whenever validation loss stopped decreasing. The loss was calculated by cross entropy. They used 200 epochs and batch size 32. Training was performed using 2 GPUs.

Images in the dataset are shaped 256×256 . The training images were resized to 230×230 . Data augmentation strategy involved randomly cropping the images to 224×224 size and randomly flipping them horizontally and vertically. The result was normalized using mean and standard deviation 0.5 each. The validation and testing images were directly resized to 224×224 and normalized.

We made some changes to this method to get the best performance possible out of our model. We used the AdamW optimizer, which works better with learning rate scheduling strategies [6]. We explored initial learning rate values in the range $[10^{-4}, 10^{-1}]$ and weight decay values in the range $[10^{-2}, 5]$. We also added brightness and contrast augmentation to improve training data. Both brightness and contrast are modified randomly by a factor in the range $[0.5, 1.5]$.

5 Implementation

We used PyTorch (1.4) [7] for implementing SatelliteNet and the training and evaluation code. Our source code is publicly available on GitHub. We used accuracy score, F-1 score and Cohen’s Kappa statistic to measure performance of SatelliteNet. We used batch size 64. Training was performed

on a Dell C4140 server with 4 NVIDIA Tesla V100 SXM2 16 GB GPUs and dual 18-core 3.70 GHz Intel Xeon Gold 6140 processors. The hardware was provided by George Washington University’s ColonialOne (Pegasus).

6 Results

We tested all 4 variants of SatelliteNet: spatial - channel, channel - spatial, spatial only and channel only, with various hyperparameters. The following results were obtained for the best model from each variant.

True	Predict						
Cloud	224	3	0	3	0	3	Accuracy 0.9390
Dust	0	187	10	1	1	3	
Haze	0	10	185	1	0	4	Cohen’s Kappa 0.9267
Land	2	5	1	195	0	2	
Seaside	0	2	1	0	198	1	F-1 score 0.9383
Smoke	3	6	9	5	0	180	

Table 3: Confusion matrix and metrics for S only model variant.

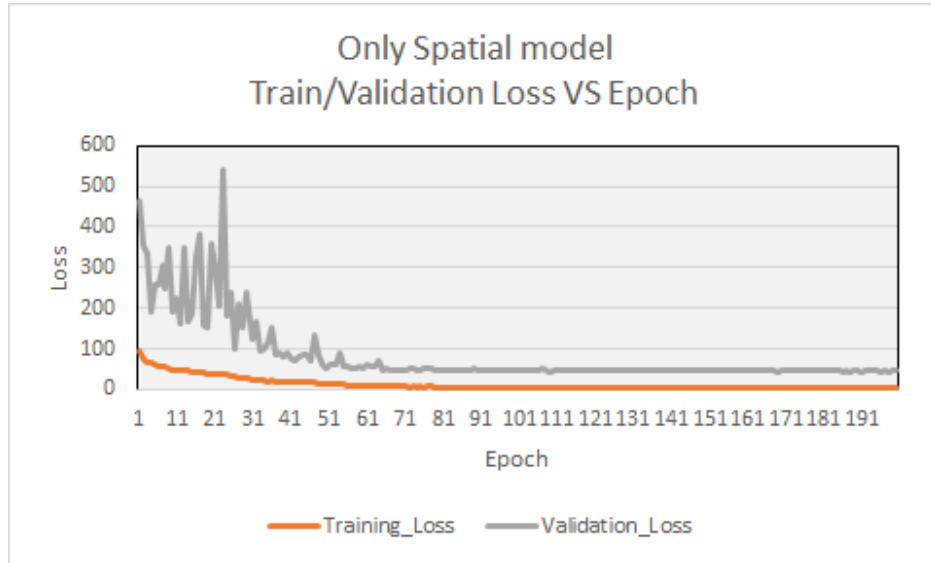


Figure 1: Training and validation loss history for the CS model variant.

True	Predict						
Cloud	220	4	1	6	0	2	Accuracy 0.9301
Dust	0	188	5	4	1	4	
Haze	0	13	178	4	0	5	Cohen's Kappa 0.9161
Land	3	6	0	193	0	3	
Seaside	0	0	4	0	198	0	F-1 score 0.9297
Smoke	4	7	9	2	0	181	

Table 4: Confusion matrix and metrics for C only model variant.

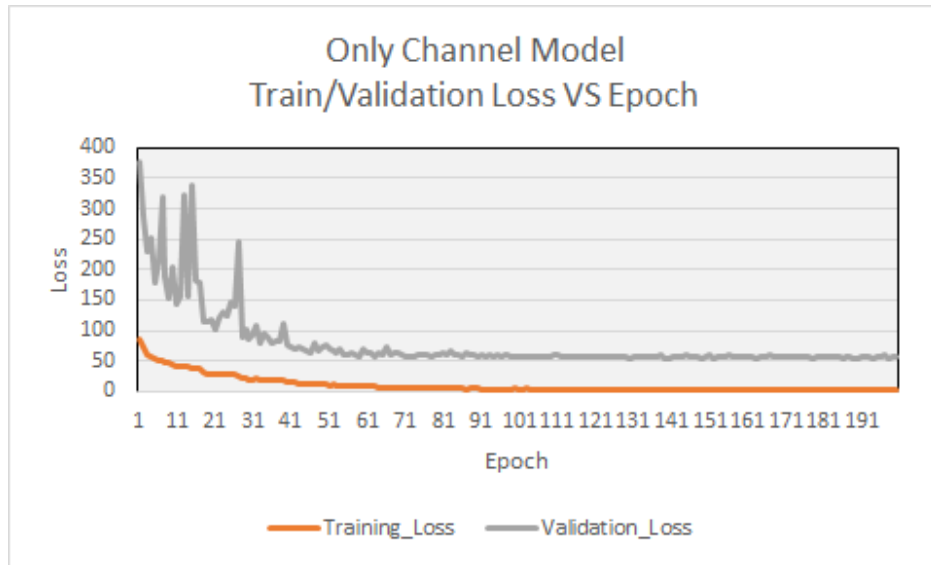


Figure 2: Training and validation loss history for the CS model variant.

True	Predict						
Cloud	222	3	0	7	0	1	Accuracy 0.9285
Dust	0	184	9	3	1	5	
Haze	0	13	175	4	0	8	Cohen's Kappa 0.9142
Land	3	5	3	188	1	5	
Seaside	0	0	3	0	199	0	F-1 score 0.9276
Smoke	2	5	4	2	2	188	

Table 5: Confusion matrix and metrics for CS model variant.

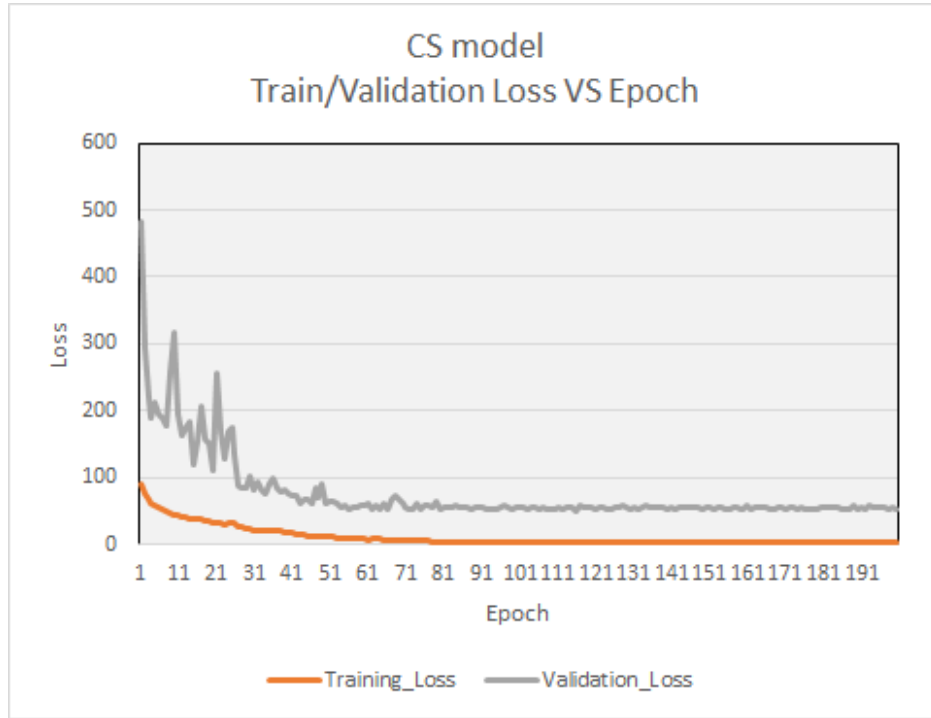


Figure 3: Training and validation loss history for the CS model variant.

True	Predict						
Cloud	221	3	0	8	0	1	Accuracy 0.9253
Dust	0	182	9	6	2	3	
Haze	0	7	186	2	0	5	Cohen's Kappa 0.9103
Land	1	4	3	194	1	2	
Seaside	0	0	4	0	194	4	F-1 score 0.9245
Smoke	5	6	11	5	1	175	

Table 6: Confusion matrix and metrics for SC model variant.

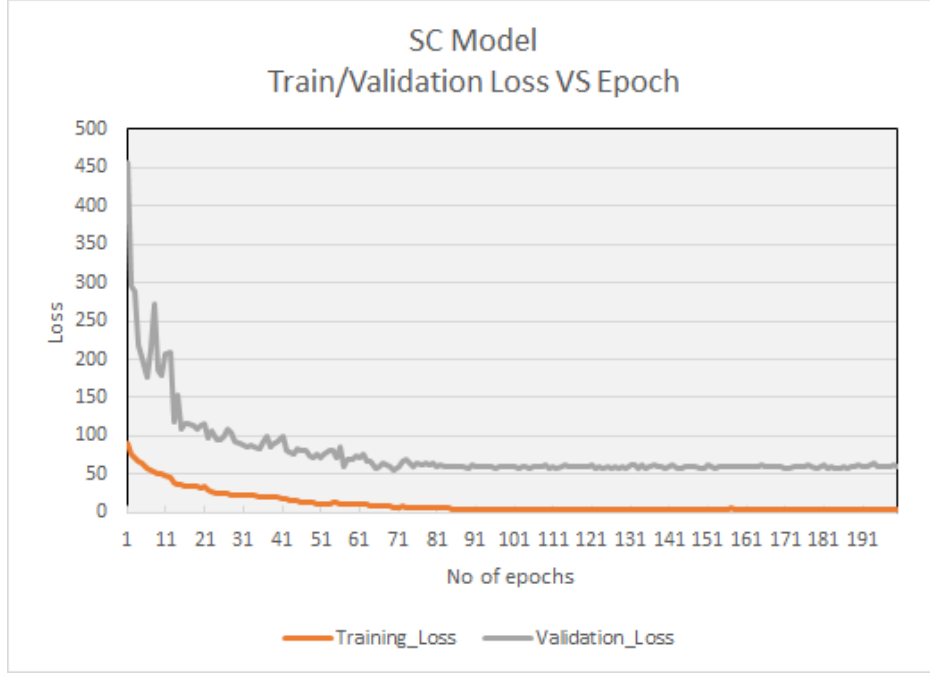


Figure 4: Training and validation loss history for the SC model variant.

The confusion matrices and associated scores clearly show that SatelliteNet performs better than SmokeNet with less number of parameters. Note that not only the overall performance of the model is better, but also the false negative rate of the Smoke class is better (smaller) for SatelliteNet (all variants except SC). The training history graphs also show that SatelliteNet finishes converging much more nearer to 100 epochs rather than 200 epochs. Which means it requires smaller amount of training time.

7 Conclusion

All variants except SC of SatelliteNet perform better than SmokeNet. We have managed to increase both accuracy by 1.15 % (from 0.9275) and Co-

hen's Kappa by 1.37 % (from 0.9130).

Furthermore, SatelliteNet is much smaller than SmokeNet in terms of trainable parameters. We cannot be sure of the exact difference since, how the fully connected layers in the blocks 1 through 4 were implemented by the authors of SmokeNet is unknown. But even if we only consider the 4th block being repeated 3 times, and count the extra convolution parameters, that gives us a difference of 7.9 M parameters.

Similar performance of all variants of the model hints that the effects of the attention mechanisms or their combinations are still being eclipsed by the complexity of the model. This could be a sign that the model size and so complexity can be reduced even more.

8 Limitations and Future Work

As we have stated before, the math was poorly described in the SmokeNet paper and we had to come up with our own interpretation of it. This is not a bad thing on its own, but it means that we cannot be sure whether the same "attention" mechanisms are being used by both SmokeNet and SatelliteNet.

The dataset put together by [1] is excellent in quality, but is still relatively small at only about 6,200 samples. A larger dataset could help improve performance and evaluation.

We don't believe that the hyperparameter space exploration was exhaustive in this study. Lack of resources prevented us from exploring more of them. There is plenty of room to further tune the model and

training process.

Another avenue for improvement is analysis of feature maps for images from different classes. This could shed some light on which attention mechanisms are better suited to the problem at hand. We leave these tasks for future work and development.

References

1. Ba, R., Chen, C., Yuan, J., Song, W. & Lo, S. SmokeNet: Satellite Smoke Scene Detection Using Convolutional Neural Network with Spatial and Channel-Wise Attention. *Remote Sensing* **11**, 1702. <https://www.mdpi.com/2072-4292/11/14/1702/pdf> (2019).
2. Chen, L. *et al.* Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 5659–5667. http://openaccess.thecvf.com/content_cvpr_2017/papers/Chen_SCA-CNN_Spatial_and_CVPR_2017_paper.pdf.
3. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 770–778. http://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf.

4. Ba, R., Chen, C., Yuan, J., Song, W. & Lo, S. *USTC_SmokeRS Dataset* (University of North Carolina at Charlotte, 2019).
<https://webpages.uncc.edu/cchen62/dataset.html>.
5. Wang, F. *et al.* *Residual attention network for image classification* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 3156–3164.
http://openaccess.thecvf.com/content_cvpr_2017/papers/Wang_Residual_Attention_Network_CVPR_2017_paper.pdf.
6. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization.
arXiv preprint arXiv:1711.05101.
<https://arxiv.org/pdf/1711.05101.pdf> (2017).
7. Paszke, A. *et al.* in *Advances in Neural Information Processing Systems 32* (eds Wallach, H. *et al.*) 8024–8035 (Curran Associates, Inc., 2019).
<http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.